

SURFACE MATCHING FOR CORRELATION OF VIRTUAL MODELS: THEORY AND APPLICATION

Roberto CARACCILO, Francesco FANTON and Alessandro GASPARETTO

DIMEG - Department of Innovation in Mechanics and Management
University of Padova

Via Venezia 1 - 35131 Padova - ITALY

Tel: ++39 49 828 6857 Fax: ++39 49 828 6816

e-mail: gaspare@hpdimeg.unipd.it

ABSTRACT

Virtual reality can enable a robot user to off line generate and test in a virtual environment a sequence of operations to be executed by the robot in an assembly cell. Virtual models of objects are to be correlated to the real entities they represent by means of a suitable transformation. A solution to the correlation problem, which is basically a problem of 3-dimensional adjusting, has been found exploiting the surface matching theory. An iterative algorithm has been developed, which matches the geometric surface representing the shape of the virtual model of an object, with a set of points measured on the surface in the real world. A peculiar feature of the algorithm is to work also if there is no one-to-one correspondence between the measured points and those representing the surface model. Furthermore, the problem of avoiding convergence to local minima is solved, by defining a starting set of states ensuring convergence to the global minimum. The developed algorithm has been tested by simulation. Finally, this paper proposes a specific application, i.e. correlating a robotized cell, equipped for biomedical use, with its virtual representation.

1. INTRODUCTION

The most recent developments of computer graphics allow to create high quality virtual representations of real entities.

Such virtual images provide a useful representation of the real world only if a transformation is defined, correlating the virtual models to the real world.

This paper investigates the problem of finding a correlation between a real entity and its virtual model. Such a problem is often encountered in many specialistic fields (i.e. in biomedical applications).

Virtual reality, intended as the capability to represent a 3-dimensional environment by means of virtual models of the objects constituting it, is used in robotics as a powerful support to off line programming.

As a matter of fact, the off line programming technique increases the productivity of a robotized cell, by avoiding that the robot be stopped for a long time, in order to be reprogrammed by means of teach-in operations.

Recent developments of CAD systems allow to build robotic simulators that can associate the typical CAD data structures with high quality images. These features enable the user of the simulator to off line generate operating sequences representing the movements of the robot and to test its interactions with

the parts inside the cell.

These sequences, easily generated in the virtual environment, can be applied to the real cell only if the correlation between the virtual and the real cell is known. A recent paper [1] describes a procedure of 2-dimensional adjusting, that finds this correlation in the case of an object lying on a working table. This procedure has been tested and applied to the field of automatic assembling. An infrared sensor is used to detect the position of the object.

This paper proposes a more general solution of the adjusting problem, i.e. a solution in the 3-dimensional case. The approach to the problem is rather different: a laser sensor has been used instead of the infrared sensor, so that analog distance measurement in a longer range are now possible, and the developed algorithm is based on the surface matching theory instead of simpler 2-dimensional geometric considerations.

The paper is organized in three main sections: the first section contains an overview of the surface matching theory, the second one proposes an *ad hoc* algorithm to solve the surface matching problem in the 3-dimensional case and some tests to validate it; finally, a combined robotic and biomedical application is discussed.

2. THEORY

The surface matching theory is aimed at finding a correlation between two different representations of the same surface.

In most applications one of these representations is obtained scanning the surface of a real object by a sensor, so that all the data are referred to the sensor's reference frame, whereas the other representation is a virtual model of the same surface, stored in the memory of a computer.

A two-level definition of the matching problem can be given, depending on how the surfaces are represented:

1) given two sets of points, representing the same surface in two different reference frames, find the rigid transformation (expressed by a rototranslation matrix) mapping one set of points into the other. Such a transformation has the following characteristics:

a) must be optimal with respect to some criterion (e.g. minimize the maximum or the mean squared difference of the distances between corresponding points);

b) must work for sets of points with different dimensions;

c) must work also if points in one set do not correspond exactly to points in the other;

d) must work also if the points of the data set are corrupted by noise.

2) given a real surface and its virtual model, a set of points is obtained by sampling the real surface. Find the rigid transformation rototranslating the modeled surface in order to minimize its "distance" from the set of points. This case may be seen as a generalization of the previous one: therefore, the transformation must have the same characteristics a) thru d).

2.1 State of the art

The following are some more formal remarks on the matching problem.

Be X a set of points and (R, \vec{t}) a rototranslation defined by a rotation matrix R and a translation vector \vec{t} , let us call P the set of points obtained applying the rototranslation (R, \vec{t}) to the set X . It is simple to obtain the rototranslation matrix (R, \vec{t}) starting from the knowledge of X and P , if the one-to-one correspondence of the points of the two sets is known. The problem of determining the transformation (R, \vec{t}) becomes more difficult if the points of one set are affected by noise, in the sense that the relationship

$$\vec{x}_i = R \cdot \vec{p}_i + \vec{t} \quad (1)$$

does not hold for all pairs of points of X and P . In the

above equation \vec{x}_i and \vec{p}_i are the coordinates of the i -th point ($i = 1 \dots N$) of the sets X and P respectively.

In this case the problem becomes a minimization problem: it is required to find the matrix R and the vector \vec{t} that minimize the sum of the errors

$$\vec{e}_i = \vec{x}_i - R \cdot \vec{p}_i - \vec{t} \quad (2)$$

The general matching problem does not require any one-to-one correspondence between the points of X and the points of P . This implies that no rototranslation exists, which maps exactly every point of X into a point of P even in the case of zero noise.

Some authors have investigated the matching problem, applying their algorithms to specific cases.

If the one-to-one correspondence is known the matching problem can be solved using the methods proposed by Horn and Haralick.

Horn [2] proposes a very simple method to determine the rototranslation matrix in the 2-dimensional case (i.e. when all points of each set lie in the same plane). A 3x4 rototranslation matrix is obtained finding the 3x3 rotation matrix first and then the 3x1 translation vector.

The algorithm proceeds as follows: given two sets of points X and P , with the same dimension N , considering three not aligned points of the P set, and the corresponding ones of the X set, build an adequate reference frame for each set of points, according to the following rules:

a) the origin be the first point;

b) the X axis be the line connecting the first and the second point;

c) the Y axis be the line, coplanar with the three points, and normal to the X axis;

d) the Z axis be chosen following the right hand rule.

Once these frames have been built, it is straightforward to find the 3x3 rotation matrix R between them. The translation vector is then found recalling that corresponding points in the two sets are linked by the following relationship:

$$\vec{x}_i = R \cdot \vec{p}_i + \vec{t} \quad (3)$$

The algorithm yields an exact result only if the points are not affected by noise; otherwise, it is not possible to find the translation satisfying the above equation exactly, but the transformation correlating each pair of corresponding points is affected by an error:

$$\vec{e}_i = \vec{x}_i - R \cdot \vec{p}_i - \vec{t} \quad (4)$$

Thus, the problem becomes: find the rigid transformation that minimizes the sum of the squared errors, due to the transformation of all the points of the

P set:

$$\varepsilon^2 = \sum_{i=1}^N \|e_i\|^2 \quad (5)$$

Another approach to solve the problem in the 2-dimensional case has been developed by Haralick [3]. His method finds the 3x3 rotation matrix considering all the points in the set simultaneously (whereas the Horn's technique considered only three points at a time). If $N = N_x = N_p$ is the number of points in each of the two sets, the mean squared error ε^2 to be minimized is now:

$$\varepsilon^2 = \sum_{i=1}^N w_i \cdot \|\bar{x}_i - (\mathbf{R} \cdot \bar{p}_i + \bar{t})\|^2 \quad (6)$$

where the weights w_i should meet the conditions:

$$w_i \geq 0, \quad \sum_{i=1}^N w_i = 1 \quad (7)$$

By choosing the weights in a convenient way, the method is made robust and stable. A good rule to choose the weights is to associate a greater weight to those points with lower squared error. The steps to build the Haralick estimator are the followings:

- a) starting from an initial value for the rotation matrix \mathbf{R} and the translation vector \bar{t} , determine the errors e_i^2 for each pair of corresponding points;
- b) the weights can now be chosen, using the Tukey function, applied to the errors e_i^2 :

$$w_i = \begin{cases} \left[1 - \frac{\|e_i\|^2}{(c \cdot S)^2} \right]^2 & \text{if } \|e_i\| \leq c \cdot S \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where c and S are parameters of the Tukey function; namely, c is chosen between 6 and 12, and S is the median of the absolute deviation of the errors e_i^2 ;

- c) solve the minimization problem using the weights that have been computed in the previous step; in this way new values for \mathbf{R} and \bar{t} are obtained;
- d) iterate the steps b) and c) until the global error ε^2 decreases below a fixed threshold.

The Haralick technique can be extended to the 3-dimensional case (see [3]).

Another solution to the surface matching problem in the 3-dimensional case is given by Besl. He proposes a method, based on the Iterative Closest Point (ICP) algorithm, to match two 3-dimensional surfaces. This technique, described in [4], utilizes quaternions to represent rotations; thus, the rototranslation

transformation is described by a 7-dimensional vector instead of a 3x4 matrix. This method reveals itself accurate and computationally efficient; furthermore, it works also if there is no one-to-one correspondence between the two sets of points representing the surfaces.

3. DESCRIPTION OF THE ALGORITHM

An algorithm has been developed, which matches the descriptor of a surface representing a virtual model (e.g. a set of points gotten from the model), with a surface descriptor extracted from the corresponding real object (e.g. a set of points measured on the surface of the object in the real world). This algorithm is a modification and an evolution of the Closest Point Algorithm proposed by Besl [4]. An important feature of this algorithm is that it works also if there is no one-to-one correspondence between the points of the X and the P sets.

Some preliminary definitions will now be given. Let us call X the model set, i.e. a set of points representing the modeled surface and P the data set, i.e. a set of points representing the real surface (e.g. points gotten sampling the surface by means of a sensor). Both sets have the same dimension N .

The matching problem is solved finding:

- a correspondence K between the two sets of points;
- a rotation matrix \mathbf{R} and a translation vector \bar{t} linking each point of the model with the corresponding data point, that minimizes the sum of the squared errors (5).

Two kinds of errors that are implicitly included in (5) are: the measurement errors (affecting the data set P), and the errors in the model (affecting the model set X). The latter are due to the fact that the virtual surface is not an exact model of the real surface; the former in most cases may be neglected. However, both these errors cannot be minimized by the matching algorithm.

The matching problem can be classified into:

- global matching
- local matching.

In the first case, there is a biunivocal correspondence between all the points of the model and all the data points, because the data represent the whole surface. It is required to determine \mathbf{R} and \bar{t} that minimize the function G :

$$G(X, P) = \min_{(\mathbf{R}, \bar{t})} \|X - (\mathbf{R} \cdot P + \bar{t})\| \quad (9)$$

In the case of local matching, the data represent only a part of the surface (thus, the dimension of P is necessarily smaller than the dimension of X). It is

required to determine not only R and \vec{t} , but also which part Z of the model X minimizes the function L :

$$L(X, P) = \min_{Z \subset X} \left[\min_{(R, \vec{t})} \|Z - (R \cdot P + \vec{t})\| \right] = \min_{Z \subset X} G(Z, P) \quad (10)$$

Before going further into the description of the algorithm, let us define which type of distance between geometric entities is assumed in the algorithm. The distance between two points $\vec{r}_1 = (x_1, y_1, z_1)$ and $\vec{r}_2 = (x_2, y_2, z_2)$ is assumed to be the euclidean distance:

$$d(\vec{r}_1, \vec{r}_2) = \|\vec{r}_1 - \vec{r}_2\| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (11)$$

Let $A = \{\vec{a}_i\}$ be a set of points, ($i=1..N_a$), where N_a is the number of points, the distance between a point \vec{p} and the set A is defined as:

$$d(\vec{p}, A) = \min_{i \in \{1..N_a\}} d(\vec{p}, \vec{a}_i) \quad (12)$$

Let l be a segment connecting \vec{r}_1 and \vec{r}_2 , the distance between a point \vec{p} and the segment l is:

$$d(\vec{p}, l) = \min_{u+v=1} \|u \cdot \vec{r}_1 + v \cdot \vec{r}_2 - \vec{p}\| \quad (13)$$

where $u \in [0, 1]$ and $v \in [0, 1]$.

Then, if $L = \{l_i\}$ is a set of segments, ($i=1..N_l$), where N_l is the number of segments, the distance between a point \vec{p} and the set L is defined as:

$$d(\vec{p}, L) = \min_{i \in \{1..N_l\}} d(\vec{p}, l_i) \quad (14)$$

Let t be a triangle whose vertices are \vec{r}_1 , \vec{r}_2 and \vec{r}_3 ; the distance between a point \vec{p} and the triangle t is defined as:

$$d(\vec{p}, t) = \min_{u+v+w=1} \|u \cdot \vec{r}_1 + v \cdot \vec{r}_2 + w \cdot \vec{r}_3 - \vec{p}\| \quad (15)$$

where $u \in [0, 1]$, $v \in [0, 1]$ and $w \in [0, 1]$.

Then, if $T = \{t_i\}$ is a set of triangles, ($i=1..N_t$), where N_t is the number of triangles, the distance between a point \vec{p} and the set T is defined as:

$$d(\vec{p}, T) = \min_{i \in \{1..N_t\}} d(\vec{p}, t_i) \quad (16)$$

Both a curve and a parametric surface are described by a relationship $\vec{r}(\vec{u})$, where:

$$\begin{aligned} \vec{u} = u \in A \subset \mathfrak{R}^1 & \quad \text{for parametric curves;} \\ \vec{u} = (u, v) \in A \subset \mathfrak{R}^2 & \quad \text{for parametric surfaces.} \end{aligned}$$

The domain A is a segment if $\vec{r}(\vec{u})$ is a curve; it is a closed region in the plane if $\vec{r}(\vec{u})$ is a surface.

The distance between a point \vec{p} and the parametric entity E is defined as:

$$d(\vec{p}, E) = \min_{\vec{r}(\vec{u}) \in E} d(\vec{p}, \vec{r}(\vec{u})) \quad (17)$$

Then, if $F = \{E_i\}$ is a set of parametric entities, ($i=1..N_e$), where N_e is the number of entities, the distance between a point \vec{p} and the set F is defined as:

$$d(\vec{p}, F) = \min_{i \in \{1..N_e\}} d(\vec{p}, \vec{E}_i) \quad (18)$$

These mathematical concepts will be useful in the following description of the global surface matching algorithm. The subscript k is used to indicate the quantities involved in the k -th iteration of the algorithm.

Let $P = \{\vec{p}_i\}$ and $X = \{\vec{x}_i\}$ be the two sets of points to be matched.

If P and X have the same dimension ($N_x = N_p$), the matching problem can be solved using the above described Haralick method, setting the initial conditions: $R_0 = I_3$, $\vec{t}_0 = \vec{0}$ (so that $P_0 \equiv P$). We define the Q operator as the function that performs the registration between P and X , i.e. computes the optimal rotation matrix that matches P and X . So, for each iteration new values for R and \vec{t} are obtained by applying the Q operator as follows:

$$(R_k, \vec{t}_k, d_k) = Q(P_k, X) \quad k > 1 \quad (19)$$

where d_k is the mean squared error given by (5). The value of P_k is obtained applying the rotation R_{k-1} and the translation \vec{t}_{k-1} to the whole set P_{k-1} , summarized by the formula:

$$P_k = R_{k-1} P_{k-1} + \vec{t}_{k-1} \quad (20)$$

The iterations stop when the absolute value of the difference between two consecutive mean squared

errors is lower than a fixed positive threshold τ :

$$|d_k - d_{k+1}| < \tau \quad (21)$$

Now, let us consider the more general problem of matching two sets of points with different dimensions. To solve this problem an iterative algorithm of the "closest point" type is used.

Let us suppose that the dimension of the set of the model points is greater than that of the set of data points ($N_x > N_p$), and let us call Y_k the set of the N_p points of X which are the closest to the points of P (i.e. are the "best correspondent points") in the k -th iteration; this defines, for each iteration, a new correspondence K . Let us call C the operator performing this computation:

$$Y_k = C(P_k, X) \quad (22)$$

Now the optimal rotation matrix R and the optimal translation vector \bar{t} can be computed using the above defined Q operator applied to the Y_k set:

$$(R_k, \bar{t}_k, d_k) = Q(P_k, Y_k) \quad (23)$$

The rototranslation (R_k, \bar{t}_k) thus computed is then applied to all the points of X , obtaining a new set P_{k+1} which is closer to the X set (see [4] for a proof). The C operator is now applied to the new set P_{k+1} in order to determine the new set Y_{k+1} of points closest to X .

The loop is iterated until the difference between the mean squared errors in two consecutive iterations is lower than a fixed positive threshold τ .

The convergence of this algorithm to a local minimum has been demonstrated [4]. However, the convergence to the global minimum is not assured in the general case. A way to make the algorithm converge to the smallest local minimum is to start the algorithm choosing R_0 in an adequate set of initial rotations, called "states", instead of choosing $R_0 = I_3$. Besl [4] has investigated how to find a suitable set of initial states.

This algorithm can be used also to solve the local matching problem; in this case, it is necessary to introduce a set of initial translations in addition to the set of initial rotations, in order to avoid convergence to a local minimum.

The algorithm has been tested for both global and local matching. The set X of points of the model is made of 55 points sampled on the surface of an ellipsoid. Four tests have been made using different P sets.

The first test applies the algorithm to the case of a global matching in ideal conditions (i.e. there is no noise). Of course, the algorithm converges exactly.

In the second test a gaussian noise has been added, to account for errors in the model and in the measurement. The error has zero mean and its variance is one twentieth of the maximum absolute value of the coordinates of the data points.

The algorithm converges after testing four initial rotations.

Fig. 1 and 2 show the sets before and after the algorithm has been run.

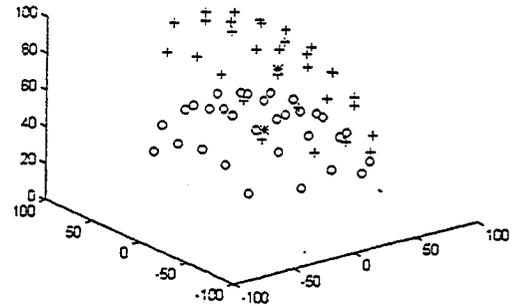


Figure 1 - Sets of points to be matched

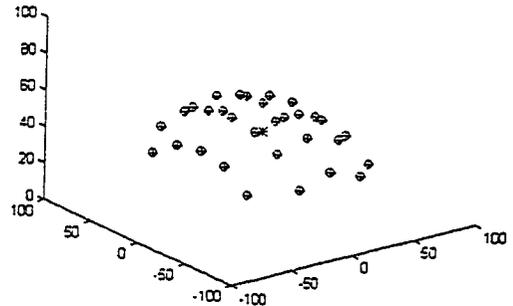


Figure 2 - Result of matching

The third test is a local matching between 24 and 55 points, with additional noise. The results are shown in Fig. 3 and 4.

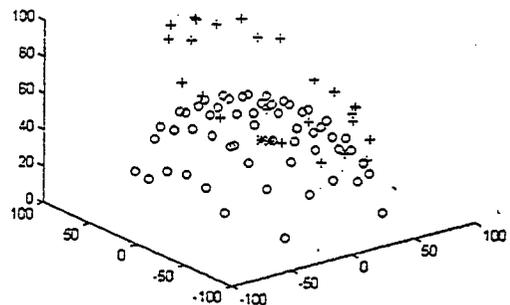


Figure 3 - Sets of points to be matched

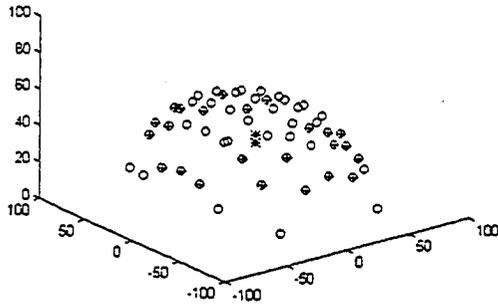


Figure 4 - Result of matching

Finally, the algorithm has been run to match two sets of 15 and 55 points, with a gaussian noise with zero mean and variance three times greater than the previous value. The results are shown in Fig. 5 and 6.

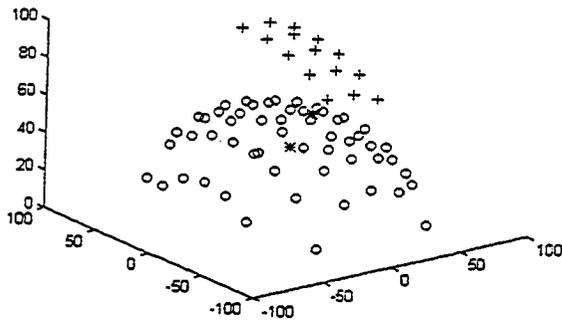


Figure 5 - Sets of points to be matched

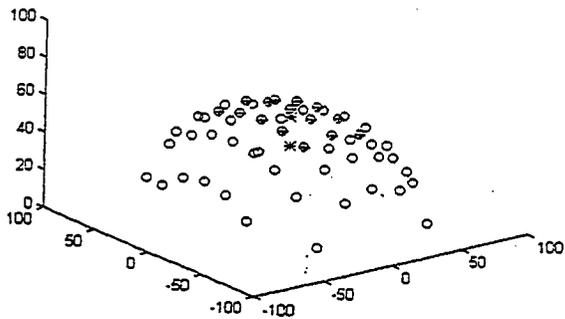


Figure 6 - Result of matching

This technique to match two sets of points can be extended to the case when the model is not represented by a set of points but by a surface. This can be convenient when the surface is expressed in an analytical form (e.g. if the model is built using a CAD system, or if the analytical form of the surface is known). To run the algorithm, the points of the model set are chosen to be the points on the surface which are the nearest to each point of the data set.

In this case the algorithm requires a method to compute the distance between a point and a parametric surface. To compute the distance between a point and a parametric surface, the latter can be approximated by a set of triangles, whose vertices lie on the surface. The shorter the edges of the triangles are, the better the approximation is.

Therefore, the problem of computing the distance between a point and a parametric surface is turned into the problem of computing the distance between a point and a set of triangles, which has been defined above.

The next three figures represent the matching between the set of the data points and the surface expressed in an analytical form.

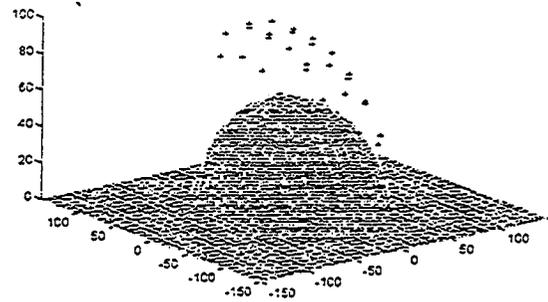


Figure 7 - The surface and the set of points to be matched

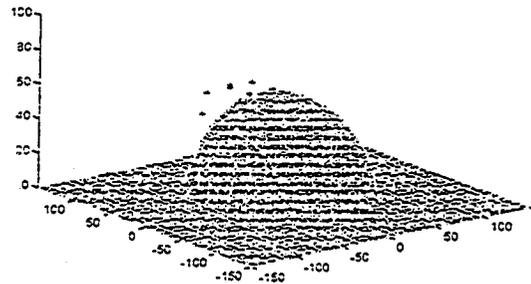


Figure 8 - An intermediate result of matching

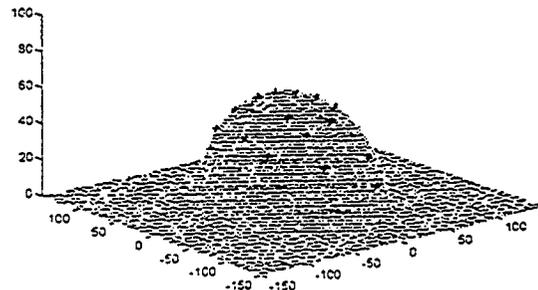


Figure 9 - Result of matching

4. APPLICATION TO ROBOTICS

A special system has been developed, whose structure is made of two levels: a real workcell equipped with an ASEA type IRB2000 industrial 6 d.o.f. robot and a simulation environment based on a prototype of a robotic simulator able to represent exactly the real cell. This system was initially conceived for automatic assembling purposes; therefore, it has been equipped with a sliding worktable and a set of automatically changeable tools (e.g. grippers, screwers, IR and laser sensors).

The flexibility of the system is such that it can be used also in a rather different research field as robotics applied to biomedics. While in the automatic assembling field a 2-dimensional adjusting procedure turned out to be sufficient in most cases, a more sophisticated 3-dimensional adjusting procedure is necessary to find the real position of the object, being this now a patient.

A major difference from the mechanical case, where all parts are modeled by means of a CAD system which is already integrated with the simulator, is that in the biomedical case the virtual models of the part of interest of the patient are obtained by correlating images gotten by different diagnostical exams (e.g. NMR, CT, DA). (See [6], [7], [10], [14], [15], [19], [20] for reference).

The proposed application refers to a skull represented by a dumb in the real world, and by a virtual model reconstructed from diagnostical images in the simulation environment.

A correlation is then to be established between the virtual model and the patient's skull, by matching the virtual surface and a set of points taken on the real skull using a laser sensor mounted on the end effector of the robot. In this way it is also possible to make a further correlation between the skull and the robot, so that to establish a full correlation between the virtual reality and the real world.

An operating procedure on this skull can then be defined in the simulation environment, and the 3-dimensional surface matching based adjusting procedure can be used to translate the operational sequence into a code suitable for the real cell, and executable by the robot. The translation procedure can be done automatically, since the translator already developed for the robotized assembling system can be employed.

5. CONCLUSIONS

The problem of finding a correlation between a real entity and its virtual model has been investigated in this paper. Solution to this problem can provide a

powerful tool in robotics, particularly useful for off line programming.

An algorithm has been proposed, based on the surface matching theory, which matches the surface of a real object with its virtual model. Two cases have been taken into account, namely the matching between two sets of points representing the real and the modeled

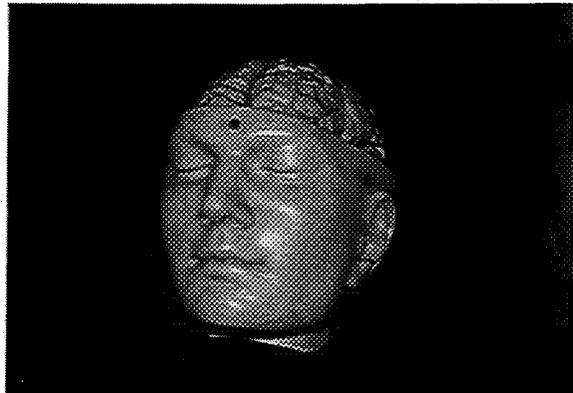


Figure 10 - The real skull

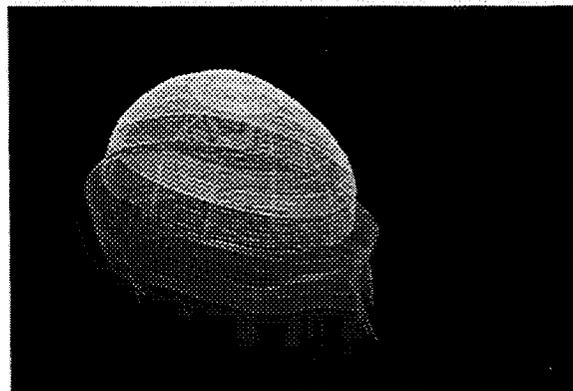


Figure 11 - The CAD model of the skull

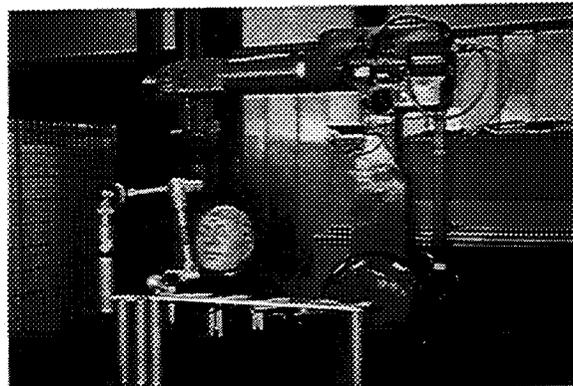


Figure 12 - The real cell

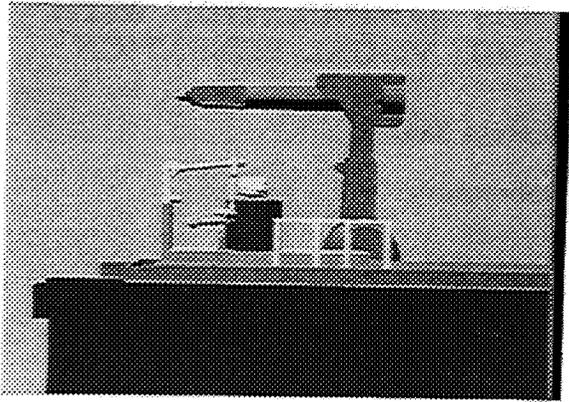


Figure 13 - The virtual model of the cell

surface respectively, and the matching between a set of points gotten from the real surface and the virtual surface, expressed in an analytical form. A peculiar feature of the algorithm is to work also if the two sets of points have different dimensions, and if there is no one-to-one correspondence between them. Moreover, both the global and the local matching problems have been defined and a solution to them has been proposed. The proposed algorithm has been tested by simulation. Finally, a special system, composed of a robotized cell and a simulation environment, initially conceived for automatic assembling purposes, has been presented, and its application to the biomedical field has been discussed.

REFERENCES

- [1] S. Badocco, R. Caracciolo, M. Giovagnoni, A. Rossi, "An On-Line Adjusting System for Assembly Robots", Proc. of the Second International Symposium on Measurement and Control in Robotics, Tsukuba Science City, Japan, November 1992.
- [2] B. K. P. Horn, "Closed Form Solution of Absolute Orientation Using Unit Quaternion", J. Opt. Soc. Am. A, vol. 4, n. 4, April 1987.
- [3] R. M. Haralick et al., "Pose Estimation from Corresponding Point Data", in Machine Vision for Inspection and Measurement, H. Freeman ed., C. E. New York Academic, 1989.
- [4] P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes", IEEE Transactions on Pattern Analysis Intelligence, Vol. 14, n. 2, February 1992.
- [5] P. J. Besl, "The Free-Form Surface Matching Problem", in Machine Vision for Three-dimensional Scenes, H. Freeman ed., C. E. New York Academic, 1990.
- [6] N. T. S. Evans, "Combining Imaging Techniques", Clin. Physiol. Meas., vol. 11, Suppl. A, pagg. 97-102, 1990.
- [7] R. Dann et al., "Evaluation of Elastic Matching System for Anatomic (CT, MR) and Functional (PET) Cerebral Images", Journal of Computer Assisted Tomography, 13(4): pagg. 603-611, July/August 1989.
- [8] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternion", J. Opt. Soc. Am. A, vol. 4, n.4, April 1987.
- [9] P. T. Fox, J. S. Perlmutter, and M. E. Raichle, "A Stereotactic Method of Anatomical Localization for Positron Emission Tomography", Journal of Computer Assisted Tomography, n.1, January/February 1985, Raven Press, New York.
- [10] T. M. Peters, J. A. Clark, et al., "Integrated Stereotaxic Imaging with CT, MR Imaging, and Digital Subtraction Angiography", Radiology, 1986; 161: 821-826.
- [11] D. J. Hawkes, D. L. G. Hill, et al., "Preliminary Work on Interpretation of SPECT Images with the Aid of Registered MR Images and MR Derived 3D Neuro-anatomical Atlas", NATO ASI Series, Vol. F60, 3D Imaging in Medicine, 1990, Edited by K.H.Hohne et al.
- [12] B. J. Erickson and C. R. Jack, "Correlation of Single Photon Emission CT with MR Image Data Using Fiducial Markers", AJNR 14:713-720, May/June 1993.
- [13] P. Clarysse, D. Gibon et al., "A Computer-Assisted System for 3-D Frameless Localization in stereotaxic MRI", IEEE Transactions on Medical Imaging, vol.10, n.4, December 1991.
- [14] D. N. Levin, "MR and PET data merge in 3-D images of brain", Diagnostic Imaging International, 28:33, January/February 1990.
- [15] C. A. Pelizzari, G. T. Y. Chen et al., "Accurate Three-Dimensional Registration of CT, PET, and/or MR Images of the brain", Journal of Computer Assisted Tomography, 13(1): 20-26, January/ February, 1989 Raven Press, Ltd., New York.
- [16] C. A. Pelizzari, G. T. Y. Chen, "Image Correlation Techniques in Radiation Therapy Treatment Planning", Computerized Medical Imaging and Graphics, vol. 13, n. 3, 235:240, 1989, Printed in the U.S.A.
- [17] B. L. Holman, R. E. Zimmerman et al., "Computer-Assisted Superimposition of Magnetic Resonance and High-Resolution Technetium-99m-HMPAO and Thallium-201 SPECT Images of the Brain", The Journal of Nuclear Medicine, vol.32, n.8, August 1991.
- [18] D. N. Levin, C. A. Pelizzari et al., "Retrospective Geometric Correlation of MR, CT, and PET Images", Radiology 1988; 169: 817-823.
- [19] W. Zhao and T. Y. Young, "A new approach to three-dimensional reconstruction with application to autoradiographic images", Computers in Medicine, vol. 13, n. 3, 1991.
- [20] M. L. Kessler, S. Pitluck et al., "Integration of Multimodality Imaging Data for Radioterapy Treatment Planning", Int. J. Radiation Oncology Biol. Phys., vol.21, 1991, pp. 1653-1667.
- [21] S. Lavallée, P. Cinquin et al., "IGOR: Image Guided Operating Robot. Metodology, medical applications, results", Innov. Tech. Biol. Med., vol.13, No 4, 1992.
- [22] D. Glauser, P. Flury, "Conception of a Robot Dedicated to Neurosurgical Operations", Proc. of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1991.
- [23] Y. S. Kwok, J. Huo et al., "A Robot with Improved Absolute Positioning Accuracy for CT Guided Stereotactic Brain Surgery", IEEE Transaction on Biomed. Eng., vol.35, 153-160, 1988.